UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/004,197	11/14/2001	Craig Nemecek	CYPR-CD01220M	1794

7590 06/06/2005

WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER	
PROCTOR, JASON SCOTT	

ART UNIT	PAPER NUMBER
2123	

DATE MAILED: 06/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/004,197	NEMECEK ET AL.	
	Examiner	Art Unit	
	Jason Proctor	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-21 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-21 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 14 November 2001 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-21 have been presented for examination. Claims 1-21 have been rejected.

Priority

1. This Application contains a claim for the benefit of priority to U.S. Provisional Application No. 60/243,708 filed 26 October 2000. The provisional application has been reviewed and priority is denied, because the provisional application does not appear to enable the claimed invention as required under 35 U.S.C. Section 112, first paragraph. See 35 U.S.C. § 119(e)(1).

For example, the provisional application contains a set of 'powerpoint-style' drawings and datasheets describing desired features for a microcontroller or a 'system-on-chip,' but this material does not appear to contain either the text description or the drawings found in the Application. In particular, no part of the provisional application appears to disclose the method steps shown in the Application at Fig. 7.

Claim Objections

2. Claim 1 is objected to because of the following informalities: the word "and" appears to be missing in line 9, as in "commands **and** requests". Appropriate correction or clarification is required.

Technology Background

In order to better facilitate a discussion of the prior art, the Examiner offers the following definitions as standard terminology for concepts well known in the art.

The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition (2000) provides the following definitions:

- **breakpoint (1) (A) (computer routine)** Pertaining to a type of instruction, instruction digit, or other condition used to interrupt or stop a computer at a particular place in a routine when manually requested. **(B) (computer routine)** A place in a routine where such an interruption occurs or can be made to occur. **(2) (software)** A point in a computer program at which execution can be suspended to permit manual or automated monitoring of program performance or results. Types include code breakpoint, data breakpoint, dynamic breakpoint, epilog breakpoint, programmable breakpoint, prolog breakpoint, static breakpoint. *Note:* A breakpoint is said to be set when both a point in the program and an event that will cause suspension of execution at that point are defined; it is said to be initiated when program execution is suspended. **(3)** A position within a pattern set where the pattern may be segmented into multiple independent bursts while still achieving predictable behavior of the device.
- **breakpoint instruction (A)** A computer instruction that causes program flow to be halted. *See also:* address stop. **(B)** A computer instruction that causes program flow to be redirected to a monitor or debugging system. *Synonym:* breakpoint halt; dynamic stop.

- **address stop** An address that, when it is encountered by a program, causes the program to halt execution. *See also:* breakpoint instruction; instruction address stop.

Microsoft Computer Dictionary, Fifth Edition (2002), provides the following definitions:

- **break**[1] *n.* 1. Interruption of a program caused by the user pressing the Break key or its equivalent.
- **break**[2] *vb.* 1. To interrupt execution at a given spot, usually for the purpose of debugging. *See also* breakpoint.
- **breakpoint** *n.* A location in a program at which execution is halted so that a programmer can examine the program's status, the contesnts of variables, and so on. A breakpoint is set and used within a debugger and is usually implemented by inserting at that point some kind of jump, call, or trap instruction that transfers control to the debugger.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. § 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 3-6 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to

one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claim 3 recites "means for determining that the microcontroller is in a sleep state" which is inadequately described by the specification. The specification (page 5, lines 24-26) does state:

> determining that the microcontroller is in the sleep state is carried out by determining if a microcontroller clock is operating and a data line from the microcontroller is in a prescribed logic state

however this teaching does not stand on its own. The specification (page 27, lines 13-15) also states:

> If the clock signals are both absent, but either data0 or data1 is low, then the gatekeeper 602 can ascertain that the microcontroller 232 is operating in a "sleep" mode.

This teaching also does not stand on its own. The disclosure fails to teach how to determine whether a clock signal is or is not absent. While it could be speculated that the absence of a clock signal timing pulse in a predefined time interval could potentially establish that a clock signal is not present, the disclosure teaches no predefined time intervals. The teachings of the disclosure do not describe how to determine if a clock signal is "absent", and therefore inadequately describe how to determine that the microcontroller is in the sleep state. A person of ordinary skill would be forced to guess how Applicants' invention was constructed in order to make and use the same.

Claims 4 and 5 directly recite the limitation "determining if the microcontroller clock is operating" which is inadequately described by the disclosure, as set forth above.

Claim 6 is rejected by virtue of its dependence.

4. Claims 16-19 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claim 16 recites a limitation including "determining if the microcontroller and the virtual microcontroller are in a sleep state" which is inadequately described by the specification. The citations noted above in the rejection of claim 3 have been considered and do not describe how Applicants' invention performs the recited determination. A person of ordinary skill would be forced to guess how Applicants' invention was constructed in order to make and use the same.

Claims 17 and 18 directly recite the limitation "determining if a microcontroller clock is operating" which is inadequately described by the disclosure, as set forth above.

Claim 19 is rejected by virtue of its dependence.

5. Claim 21 is rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claim 21 directly recites the limitation "determining if a microcontroller clock is operating" which has been discussed above. A person of

ordinary skill would be forced to guess how Applicants' invention was constructed in order to make and use the same.

The following is a quotation of the second paragraph of 35 U.S.C. § 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 3-6 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The "means for" limitation of claim 3 appears to seek coverage under 35 U.S.C. § 112, sixth paragraph. However, as this limitation is inadequately described by the disclosure, as set forth above, it is impossible to determine the metes and bounds of a "means for" limitation relying on those teachings. As it is unknown how Applicants' invention determines that a microcontroller clock is not active, critical to determining whether the microcontroller is in a sleep mode, it is therefore unknown particularly which "means for determining" Applicant seeks to patent with claim 3.

7. Claims 4 and 5 stand rejected by virtue of their dependency, however the Examiner respectfully observes that claims 4 and 5 recite the acts for achieving the specified function of "determining that the microcontroller is in the sleep state", as taught by the disclosure, and therefore claims 4 and 5 are not properly subject to coverage under 35 U.S.C. § 112, sixth paragraph. Please see MPEP 2181. The Examiner respectfully suggests amending claims 4 and 5 such that they do not depend from claim 3, as the inventions recited by claims 4 and 5 essentially render the limitations of claim

3 meaningless. When considering the invention of claim 5, no patentable weight can be given to the limitations of claim 3, from which claim 5 depends, because claim 5 negates the 35 U.S.C. § 112, sixth paragraph, coverage of claim 3 by defining the necessary acts for achieving the "means for" functionality of claim 3.

Claims not specifically mentioned stand rejected by virtue of their dependency.

Claim Interpretation

In the interest of compact prosecution, the Examiner makes the following claim interpretations in order to apply prior art to the claims. See *Ex parte Ionescu*, 222 USPQ 537 (Bd. Pat. App. & Inter. 1984).

Claim 1 is interpreted with the language "so that halt commands **and** requests for data".

Claims 3 is interpreted as including the limitations "wherein the gatekeeper circuit comprises means known in the art for determining that the microcontroller is in a sleep state".

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-21 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 5,911,059 to Profit, Jr. (Profit) in view of "DEBUG" as described in "A DEBUG Tutorial" by Daniel B. Sedory, copyright 2004 (Sedory) and further in view of "Microsoft PressPass – Microsoft Files Summary Judgement Motions" by Microsoft® published February 12, 1999 (Microsoft).

9. Regarding claim 1, Profit teaches an in-circuit emulation system comprising:

a processor having a microcontroller clock (Fig. 7, reference 204; column 6, lines 5-24; regarding a clock in the target program in the processor, column 12, lines 24-28);

a virtual processor (referred to as a *processor model shell 212*) (column 6, lines 25-48) operating in lock step synchronization with the processor (column 11, lines 40-43);

a gatekeeper circuit (referred to as *RUN/HALT controller 240*) coupled to the virtual processor and the processor (Fig. 8, reference 240; column 8, line 65 – column 10, line 31); and

a host computer running in-circuit emulation debug software (Fig. 7, reference 214; column 6, lines 49-60) in communication with the gatekeeper circuit so that halt commands are passed through and regulated by the gatekeeper circuit (Fig. 7, reference 222; column 9, lines 4-6).

Official notice is taken that the term *microcontroller* refers to a single unit usually comprising central processing unit, memory, and I/O ports. As Profit teaches an emulator unit that contains at least these features (Fig. 7, reference 202), it would have

been obvious to a person of ordinary skill in the art at the time of Applicants' invention that Profit's emulator is readily adaptable to accept microcontrollers, as would be desired by a person whose goal it is to develop and debug code for microcontrollers.

Profit does not explicitly recite that requests for data from the virtual processor (to the actual processor) are passed through the gatekeeper circuit, however Profit does explicitly teach the use of standard software debugging tools (column 6, lines 49-60).

The Sedory reference describes the DEBUG command of Microsoft® MS-DOS® operating system versions 5.0 and later (Sedory, page 1). The Microsoft reference establishes the release date of MS-DOS® 5.0 as June 1991 (Microsoft, page 4). Therefore the Sedory reference is relied upon as describing the DEBUG command of MS-DOS® 5.0 as it was known in the art in June 1991.

Sedory teaches that the DEBUG command includes the capability of viewing memory contents (page 3, Dump command), modifying memory contents (pages 4-5, Enter command; page 8, Move command), viewing registers (pages 9-10, Register command), and modifying registers (pages 9-10, Register command). Additionally, these concepts are well known in the art of debugging as standard techniques, commonly referred to as traces, tracing, memory dumping, writing to memory locations, et cetera.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention in combination with his own knowledge of the particular art, at the explicit suggestion of Profit to combine software debugging tools, to incorporate the well known debugging techniques embodied in DEBUG with the in-circuit emulation system

taught by Profit. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to make the requisite modifications to the features of DEBUG to accommodate the communicative nature of Profit's in-circuit emulation system. This combination would render obvious the recited limitations of communicating "halt commands and requests for data from the virtual computer" through the gatekeeper circuit taught by Profit.

Regarding claim 2, Profit teaches a gatekeeper clock running independent of the microcontroller clock to clock operations carried out in the gatekeeper circuit (column 10, lines 38-41, *"In this embodiment, the simulation time keeper circuit 232 includes a* counter"; column 10, lines 44-45, *"The counter is driven by the clock signal on line 242")*.

Regarding claim 3, Profit teaches that the gatekeeper circuit comprises means for halting the microcontroller, functionally equivalent to placing it in a "sleep state" (column 8, line 65 – column 10, line 31; column 10, line 32 – column 11, line 7; especially column 9, lines 41-46). When placing a microcontroller into a sleep state, the gatekeeper circuit is reasonably apprised that the microcontroller is in a sleep state, obviating the purpose of a specialized sleep state detection ability.

Regarding claims 4 and 5, Profit teaches that the gatekeeper circuit (*RUN/HALT controller 240*) monitors the state of the microcontroller (column 9, lines 47-55). In

another embodiment, Profit teaches that the target bus watch circuit 224 (which comprises, among other components, the *RUN/HALT controller 240*) monitors "the data address and status lines on the target bus 208 of the processor emulator 202" (column 10, lines 4-6). It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, in combination with his own knowledge of the particular art as well as Profit's explicit teaching of the advantages of various embodiments, to combine and modify the teachings of Profit to arrive at the claimed invention.

Regarding claim 6, Profit teaches that the gatekeeper circuit notifies the host computer of the status of the microcontroller (column 10, lines 4-23).

Regarding claim 7, Profit teaches sending a halt command (referred to as *HOST INTERRUPT request*) to the microcontroller (referred to as *processor emulator 202* which includes *processor 204* of Fig. 7) that is received from the virtual microcontroller (referred to as *hardware simulator 210* which includes *processor model shell 212* of Fig. 7) and halting the microcontroller or virtual microcontroller in response to the halt command (column 8, lines 13-16; column 9, lines 40-61; column 10, lines 11-18).

Regarding claim 8, Profit teaches that the gatekeeper circuit (*RUN/HALT controller 240*) monitors the state of the microcontroller (column 9, lines 47-55). In another embodiment, Profit teaches that the target bus watch circuit 224 (which comprises, among other components, the *RUN/HALT controller 240*) monitors "the data

address and status lines on the target bus 208 of the processor emulator 202" (column 10, lines 4-6). Profit teaches that one of the tasks of the *RUN/HALT controller 240* is to halt the microcontroller and "communicate required event information between the processor emulator 202 and the hardware simulator 210" (column 10, lines 4-18). Profit therefore teaches detecting that a halt has occurred in the microcontroller and notifying the host computer that a break has occurred.

Regarding claims 9 and 10, the recited limitations are equivalent to a *break* (as defined by Microsoft Computer Dictionary, Fifth Edition) in claim 9 and a *breakpoint* or *breakpoint instruction* (as defined by The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition) in claim 10. Profit explicitly teaches the inclusion of standard software debugging tools (column 6, lines 49-60) which a person of ordinary skill in the art at the time of Applicants' invention would recognize as including both a *break* as well as a *breakpoint* or *breakpoint instruction.* It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, at the explicit suggestion of Profit, to include well-known and basic concept of debugging in the in-circuit emulation system taught by Profit. Regarding the recitation of a "breakpoint controller" (claim 10), Profit teaches a *RUN/HALT controller 240* that is functionally equivalent (column 10, lines 4-23).

Regarding claim 11, the recited limitations are equivalent to a *break* (as defined by Microsoft Computer Dictionary, Fifth Edition) in claim 9 and a *breakpoint* or

breakpoint instruction (as defined by The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition) in claim 10. Profit explicitly teaches the inclusion of standard software debugging tools (column 6, lines 49-60) which a person of ordinary skill in the art at the time of Applicants' invention would recognize as including both a *break* as well as a *breakpoint* or *breakpoint instruction*. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, at the explicit suggestion of Profit, to include well-known and basic concept of debugging in the in-circuit emulation system taught by Profit.

Further, the combination formed in the rejection of claim 1 teaches the limitations of "permitting access to registers and memory locations". Specifically, Sedory teaches that the DEBUG command includes the capability of viewing memory contents (page 3, Dump command), modifying memory contents (pages 4-5, Enter command; page 8, Move command), viewing registers (pages 9-10, Register command), and modifying registers (pages 9-10, Register command).

Regarding claim 12, the recited limitations are equivalent to a *break* (as defined by Microsoft Computer Dictionary, Fifth Edition). Profit explicitly teaches the inclusion of standard software debugging tools (column 6, lines 49-60), which a person of ordinary skill in the art at the time of Applicants' invention would recognize as including a *break*. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, at the explicit suggestion of Profit, to include well-known and basic concept of debugging in the in-circuit emulation system taught by Profit.

10. Claim 13 recites the method employed by the system of claims 1. To that effect, Profit teaches an in-circuit emulation system and accompanying method comprising:

a virtual processor (referred to as a *processor model shell 212*) (column 6, lines 25-48) operating in lock step synchronization with a processor (column 11, lines 40-43);

a gatekeeper circuit (referred to as *RUN/HALT controller 240*) coupled to the virtual processor and the processor (Fig. 8, reference 240; column 8, line 65 – column 10, line 31); and

a host computer running in-circuit emulation debug software (Fig. 7, reference 214; column 6, lines 49-60) in communication with the gatekeeper circuit so that halt commands are passed through and regulated by the gatekeeper circuit (Fig. 7, reference 222; column 9, lines 4-6).

Official notice is taken that the term *microcontroller* refers to a single unit usually comprising central processing unit, memory, and I/O ports. As Profit teaches an emulator unit that contains at least these features (Fig. 7, reference 202), it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention that Profit's emulator is readily adaptable to accept microcontrollers, as would be desired by a person whose goal it is to develop and debug code for microcontrollers.

Profit does not explicitly recite that requests for data from the virtual processor (to the actual processor) are passed through the gatekeeper circuit, however Profit does explicitly teach the use of standard software debugging tools (column 6, lines 49-60).

The Sedory reference describes the DEBUG command of Microsoft® MS-DOS® operating system versions 5.0 and later (Sedory, page 1). The Microsoft reference establishes the release date of MS-DOS® 5.0 as June 1991 (Microsoft, page 4). Therefore the Sedory reference is relied upon as describing the DEBUG command of MS-DOS® 5.0 as it was known in the art in June 1991.

Sedory teaches that the DEBUG command includes the capability of viewing memory contents (page 3, Dump command), modifying memory contents (pages 4-5, Enter command; page 8, Move command), viewing registers (pages 9-10, Register command), and modifying registers (pages 9-10, Register command). Additionally, these concepts are well known in the art of debugging as standard techniques, commonly referred to as traces, tracing, memory dumping, writing to memory locations, et cetera.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention in combination with his own knowledge of the particular art, at the explicit suggestion of Profit to combine software debugging tools, to incorporate the well known debugging techniques embodied in DEBUG with the in-circuit emulation system taught by Profit. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to make the requisite modifications to the features of DEBUG to accommodate the communicative nature of Profit's in-circuit emulation

system. This combination would render obvious the recited limitations of communicating "halt commands and requests for data from the virtual computer" through the gatekeeper circuit taught by Profit.

Regarding claims 14 and 15, the recited limitations are equivalent to a *break* (as defined by Microsoft Computer Dictionary, Fifth Edition) in claim 9 and a *breakpoint* or *breakpoint instruction* (as defined by The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition) in claim 10. Profit explicitly teaches the inclusion of standard software debugging tools (column 6, lines 49-60) which a person of ordinary skill in the art at the time of Applicants' invention would recognize as including both a *break* as well as a *breakpoint* or *breakpoint instruction*. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, at the explicit suggestion of Profit, to include well-known and basic concept of debugging in the in-circuit emulation system taught by Profit. Regarding the recitation of a "breakpoint controller" (claim 10), Profit teaches a *RUN/HALT controller 240* that is functionally equivalent (column 10, lines 4-23).

Regarding claim 16, Profit teaches that the gatekeeper circuit comprises means for halting the microcontroller, functionally equivalent to placing it in a "sleep state" (column 8, line 65 – column 10, line 31; column 10, line 32 – column 11, line 7; especially column 9, lines 41-46). When placing a microcontroller into a sleep state, the

gatekeeper circuit is reasonably apprised that the microcontroller is in a sleep state, obviating the purpose of a specialized sleep state detection ability.

Regarding claims 17 and 18, Profit teaches that the gatekeeper circuit (*RUN/HALT controller 240*) monitors the state of the microcontroller (column 9, lines 47-55). In another embodiment, Profit teaches that the target bus watch circuit 224 (which comprises, among other components, the *RUN/HALT controller 240*) monitors "the data address and status lines on the target bus 208 of the processor emulator 202" (column 10, lines 4-6). It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, in combination with his own knowledge of the particular art as well as Profit's explicit teaching of the advantages of various embodiments, to combine and modify the teachings of Profit to arrive at the claimed invention.

Regarding claims 19 and 20, Profit teaches that the gatekeeper circuit (*RUN/HALT controller 240*) monitors the state of the microcontroller (column 9, lines 47-55). In another embodiment, Profit teaches that the target bus watch circuit 224 (which comprises, among other components, the *RUN/HALT controller 240*) monitors "the data address and status lines on the target bus 208 of the processor emulator 202" (column 10, lines 4-6). Profit teaches that one of the tasks of the *RUN/HALT controller 240* is to halt the microcontroller and "communicate required event information between the processor emulator 202 and the hardware simulator 210" (column 10, lines 4-18). Profit

therefore teaches notifying the host computer of the state of the microcontroller and virtual microcontroller, whether that state is halted, sleep, or otherwise.

11. Claim 21 recites the method employed by a system combining the limitations of claims 5, 6, 8, 9, 10, and 11. As Profit in view of DEBUG renders all of these limitations obvious, Profit in view of DEBUG similarly renders the combination of these limitations obvious. Profit in view of DEBUG teaches the system and its operation, thereby rendering the method of its use obvious. It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Profit in view of DEBUG in the combination recited by claim 21 in order to achieve the best features of the prior art. Motivation to do so would be found in the knowledge of a person of ordinary skill in the art.

Conclusion

Art considered pertinent by the examiner but not applied has been cited on form PTO-892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin J Teska can be reached on (571) 272-3716. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3713.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

jsp

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER